

CLAIMS:

1. An apparatus for implementing a Floating-Point related application, comprising:
 2. a tool that includes:
 3. a receiver for receiving a list of commands in a computer language; the language defining Floating-Point events of interest in respect of at least one FP instruction;
 4. a parser for parsing the commands;
 5. a processor configured to process at least the parsed commands for realizing the floating-point related application on the basis of said events.
 6. 2. The apparatus of Claim 1, wherein said language further defining regrouping of the events into at least one coverage model; and wherein said processor is configured to process the parsed commands for realizing the floating-point related application on the basis of said events and said at least one coverage model.
 7. 3. The apparatus according to Claim 1, wherein said application is an evaluation of coverage of tests being run on a design.
 8. 4. The apparatus according to Claim 1, wherein said processor is configured to generate a sequence of test vectors for verification of Floating-Point module operation; the test vectors meet the constraints of said events.
 9. 5. The apparatus according to Claim 4, wherein said verification includes verifying if the Floating-Point operation complies with IEEE standard for Floating-Point.
 10. 6. For use with the Floating-Point module of Claim 1, a computer language; the language defining Floating-Point events of interest in respect of at least one FP instruction.

DETAILED DESCRIPTION

1 7. The computer language of Claim 6, further defining
2 regrouping of the events into at least one coverage model.

1 8. An apparatus for implementing a Floating-Point
2 related application, comprising:

3 a tool that includes:

4 a receiver for receiving a list of commands in a
5 computer language; the language defining Floating-Point
6 events of interest and regrouping of events into at least
7 one coverage model, in respect of at least one FP
8 instruction; the coverage model having the form of a
9 sequence of Floating-Point commands with constraints on
10 (i) at least one intermediate result operand of the FP
11 instruction, and (ii) result operand of the FP
12 instruction;

13 a parser for parsing the commands;

14 a processor for processing at least the parsed
15 commands for realizing the Floating-point related
16 application at least on the basis of said events and
17 said at least one coverage model.

1 9. The apparatus according to Claim 8, wherein said
2 application is an evaluation of coverage of tests being
3 run on a design.

1 10. The apparatus according to Claim 8, wherein said
2 processor is configured to generate a sequence of test
3 vectors for verification of Floating-Point module
4 operation; the test vectors meet the constraints of said
5 events and the at least one coverage model.

1 11. The apparatus according to Claim 8, wherein said
2 verification includes verifying if the Floating-Point
3 operation complies with the IEEE standard for
4 Floating-Point.

1 12. For use with the Floating-Point module of Claim 8, a
2 computer language; the language defining Floating-Point
3 events of interest and regrouping of events into at least

8 one coverage model, in respect of at least one FP
9 instruction, the coverage model having the form of a
10 sequence of Floating-Point commands with constraints on
11 (i) at least one intermediate result operand of the FP
12 instruction, and (ii) result operand of the FP
13 instruction.

1 **13.12.** An apparatus for implementing a Floating-Point |
2 related application, comprising:

3 a tool that includes:

4 a receiver for receiving a list of commands in a
5 computer language; the language defining Floating-Point
6 events of interest and regrouping of events into at least
7 one coverage model, in respect of at least one FP
8 instruction; the coverage model having the form of a
9 sequence of Floating-Point commands with constraints on
10 (i) at least one intermediate result operand of the FP
11 instruction, and (ii) result operand of the FP
12 instruction; each one of said constraints is expressed as
13 at least one set each of which defining allowable
14 Floating-Point numbers;

15 a parser for parsing the commands;

16 a processor for processing at least the parsed
17 commands for realizing at least on the basis of said
18 events and said at least one coverage model the Floating
19 -point related application.

1 **14.** The apparatus according to Claim 13, wherein said
2 application is an evaluation of coverage of tests being
3 run on a design.

1 **15.** The apparatus according to Claim 13, wherein said
2 processor is configured to generate a sequence of test
3 vectors for verification of Floating-Point module
4 operation; the test vectors meet the constraints of said
5 events and at least one coverage model.

1 **16.** The apparatus according to Claim 13, wherein said
2 verification includes verifying if the Floating-Point

operation complies with the IEEE standard for
3 Floating-Point.
4

17. For use with the Floating-Point module of Claim 13,
1 a computer language; the language defining Floating-Point
2 events of interest and regrouping of events into at least
3 one coverage model, in respect of at least one FP
4 instruction; the coverage model having the form of a
5 sequence of Floating-Point commands with constraints on:
6 (i) at least one intermediate result operand of the FP
7 instruction, and (ii) result operand of the FP
8 instruction; each one of said constraints is expressed as
9 at least one set each of which defining allowable
10 Floating-Point numbers.

18. The apparatus according to Claim 1, wherein said list
1 of commands includes: Range of FP numbers ; Mask on bits
2 of FP number; Set or Reset Number of Bits in an FP
3 number; Set or Reset Continuous-Bit-Long in an FP number;
4 Relative Values of at least two FP numbers, and logical
5 operations among said commands.

19. The apparatus according to Claim 6, wherein said list
1 of commands includes: Range of FP numbers ; Mask on bits
2 of FP number; Set or Reset Number of Bits in an FP
3 number; Set or Reset Continuous-Bit-Long in an FP number;
4 Relative Values of at least two FP numbers, and logical
5 operations among said commands.

20. The apparatus according to Claim 7, wherein said list
1 of commands includes: Range of FP numbers ; Mask on bits
2 of FP number; Set or Reset Number of Bits in an FP
3 number; Set or Reset Continuous-Bit-Long in an FP number;
4 Relative Values of at least two FP numbers, and logical
5 operations among said commands.

21. The apparatus according to Claim 8, wherein said list
1 of commands includes: Range of FP numbers ; Mask on bits
2 of FP number; Set or Reset Number of Bits in an FP
3 number; Set or Reset Continuous-Bit-Long in an FP number;

- 5 Relative Values of at least two FP numbers, and logical
6 operations among said commands.
- 1 **22.** The apparatus according to Claim 12, wherein said
2 list of commands includes: Range of FP numbers ; Mask on
3 bits of FP number; Set or Reset Number of Bits in an FP
4 number; Set or Reset Continuous-Bit-Long in an FP number;
5 Relative Values of at least two FP numbers, and logical
6 operations among said commands.
- 1 **23.** The apparatus according to Claim 13, wherein said
2 list of commands includes: Range of FP numbers ; Mask on
3 bits of FP number; Set or Reset Number of Bits in an FP
4 number; Set or Reset Continuous-Bit-Long in an FP number;
5 Relative Values of at least two FP numbers, and logical
6 operations among said commands.
- 1 **24.** The apparatus according to Claim 17, wherein said
2 list of commands includes: Range of FP numbers ; Mask on
3 bits of FP number; Set or Reset Number of Bits in an FP
4 number; Set or Reset Continuous-Bit-Long in an FP number;
5 Relative Values of at least two FP numbers, and logical
6 operations among said commands.
- 1 **25.** The apparatus according to Claim 8, wherein said
2 constraints are further applied to attributes of Machine
3 State.
- 1 **26.** The apparatus according to Claim 13, wherein said
2 constraints are further applied to attributes of Machine
3 State.
- 1 **27.** A method for implementing a Floating-Point related
2 application that includes the steps of :
3 (i) receiving a list of commands in a computer
4 language; the language defining Floating-Point
5 events of interest in respect of at least one FP
6 instruction;
7 (ii) parsing the commands; and

(iii) processing at least the parsed commands for realizing the floating-point related application on the basis of said events.

28. A method for implementing a Floating-Point related application that includes the steps of:

(i) receiving a list of commands in a computer language; the language defining Floating-Point events of interest and regrouping of events into at least one coverage model, in respect of at least one FP instruction; the coverage model having the form of a sequence of Floating-Point commands with constraints on (i) at least one intermediate result operand of the FP instruction, and (ii) result operand of the FP instruction; and

- (ii) parsing the commands, and
- (iii) processing at least the parsed commands for realizing the Floating-point related application at least on the basis of said events and said at least one coverage model.

29. A method for implementing a Floating-Point related application, that includes the step of:

(i) receiving a list of commands in a computer language; the language defining Floating-Point events of interest and regrouping of events into at least one coverage model, in respect of at least one FP instruction; the coverage model having the form of a sequence of Floating-Point commands with constraints on (i) at least one intermediate result operand of the FP instruction, and (ii) result operand of the FP instruction; each one of said constraints is expressed as at least one set each of which defining allowable Floating-Point numbers;

(ii) parsing the commands; and

(iii) processing at least the parsed commands for realizing at least on the basis of said events and said at least one coverage model the Floating-point related application.

30. A program storage device readable by machine, tangibly embodying a program of instructions executable by the machine to perform method steps for implementing a Floating-Point related application that includes the steps of :

(i) receiving a list of commands in a computer language; the language defining Floating-Point events of interest in respect of at least one FP instruction;

(ii) parsing the commands; and

(iii) processing at least the parsed commands for realizing the floating-point related application on the basis of said events.

31. A computer program product comprising a computer useable medium having computer readable program code embodied therein for causing the computer to implement a Floating-Point related application, comprising:

computer readable program code for causing the computer to receive a list of commands in a computer language; the language defining Floating-Point events of interest in respect of at least one FP instruction;

computer readable program code for causing the computer to parse the commands; and

computer readable program code

computer readable program code for causing the computer to process at least the parsed commands for realizing the floating-point related application on the basis of said events.

..... program storage device readable by machine,

32. A program storage device readable by machine tangibly embodying a program of instructions executable by the machine to perform method steps for implementing a

4 Floating-Point related application, that includes the
5 steps of:

6 (i) receiving a list of commands in a computer
7 language; the language defining Floating-Point
8 events of interest and regrouping of events into
9 at least one coverage model, in respect of at
10 least one FP instruction; the coverage model
11 having the form of a sequence of Floating-Point
12 commands with constraints on (i) at least one
13 intermediate result operand of the FP instruction,
14 and (ii) result operand of the FP instruction;
15 each one of said constraints is expressed as at
16 least one set each of which defining allowable
17 Floating-Point numbers;

18 (ii) parsing the commands; and

19 (iii) processing at least the parsed commands for
20 realizing at least on the basis of said events and
21 said at least one coverage model the Floating
22 -point related application.

1 33. A computer program product comprising a computer
2 useable medium having computer readable program code
3 embodied therein for causing the computer to implement a
4 Floating-Point related application, comprising:
5 computer readable program code for causing the
6 computer to receive a list of commands in a computer
7 language; the language defining Floating-Point events of
8 interest and regrouping of events into at least one
9 coverage model, in respect of at least one FP instruction;
10 the coverage model having the form of a sequence of
11 Floating-Point commands with constraints on (i) at least
12 one intermediate result operand of the FP instruction, and
13 (ii) result operand of the FP instruction; each one of
14 said constraints is expressed as at least one set each of
15 which defining allowable Floating-Point numbers;

computer readable program code for causing the computer to parse the commands; and

computer readable program code for causing the computer to process at least the parsed commands for realizing at least on the basis of said events and said at least one coverage model the Floating-point related application.

34. A program storage device readable by machine, tangibly embodying a program of instructions executable by the machine to perform method steps for implementing a Floating-Point related application, that includes the steps of:

(i) receiving a list of commands in a computer language; the language defining Floating-Point events of interest and regrouping of events into at least one coverage model, in respect of at least one FP instruction; the coverage model having the form of a sequence of Floating-Point commands with constraints on (i) at least one intermediate result operand of the FP instruction, and (ii) result operand of the FP instruction; each one of said constraints is expressed as at least one set each of which defining allowable Floating-Point numbers;

- (ii) parsing the commands; and
- (iii) processing at least the parsed commands for realizing at least on the basis of said events and said at least one coverage model the Floating-point related application.

-point related application.

35. A computer program product comprising a computer useable medium having computer readable program code embodied therein for causing the computer to implement a Floating-Point related application, comprising:
computer readable program code for causing the computer to receive a list of commands in a computer

7 language; the language defining Floating-Point events of
8 interest and regrouping of events into at least one
9 coverage model, in respect of at least one FP instruction;
10 the coverage model having the form of a sequence of
11 Floating-Point commands with constraints on (i) at least
12 one intermediate result operand of the FP instruction, and
13 (ii) result operand of the FP instruction; each one of
14 said constraints is expressed as at least one set each of
15 which defining allowable Floating-Point numbers;

16 computer readable program code for causing the
17 computer to parse the commands; and

18 computer readable program code for causing the
19 computer to process at least the parsed commands for
20 realizing at least on the basis of said events and said
21 at least one coverage model the Floating -point related
22 application.

DRAFT COPY
DO NOT DISTRIBUTE